JOURNAL OF SOFTWARE MAINTENANCE: RESEARCH AND PRACTICE
*J. Softw. Maint: Res. Pract.* 2000; **12**:47–78

**Research**

# Measure-driven processes and architecture for the empirical evaluation of software technology

Giovanni Cantone*,†

*Department of Informatics, Systems, and Production Management Engineering, University of Rome at Tor Vergata, Roma, Italy*

**SUMMARY**

**This paper reports on an on-going project undertaken within the University of Rome at Tor Vergata to automate the business processes of a class of administrative organizations. The experience gained and the lessons learned, while performing in the role of Innovation-driver, are abstracted and presented. In the context of this experience with process re-engineering, the paper compares the utility of empirically investigating software technology transfer with the reuse of technology measurement models possibly available. The paper also discusses the roles played by various environmental factors—technology, experimentation and measurements, an organization's business objectives, and other organizational issues, including the expected benefits and costs—in the empirical investigation of software technology transfer. The paper concludes by presenting a short case study and three lessons learned. Copyright © 2000 John Wiley & Sons, Ltd.**

KEY WORDS: software maintenance processes; process reengineering; technology transfer; maintenance metrics; empirical research; experimental software engineering; conceptual architecture

## 1. INTRODUCTION

Software (SW) technology may often have a hard rather than a soft *impact* on organizations. In fact, SW technology can change its range of influence from introducing or switching to a new tool, to radically re-engineering the organization's business processes. Moreover, SW technology carries a

*Correspondence to: Dr. Giovanni Cantone, Department of Informatics, Systems, and Production Management Engineering, University of Rome at Tor Vergata, S.N.C. Roma, Italy.
†Email: cantone@info.uniroma2.it

Contract/grant sponsor: University of Rome at Tor Vergata: International Cooperation Projects, Grant # (UKaiserslautern, DE-UMD. USA-URM2, I)/1997-99/1.

*Received 10 February 1999*
*Revised 20 October 1999*

Table I. Definitions of the abbreviations used in this paper.

| Abbreviation | Definition |
|---|---|
| BPR | business process re-engineering |
| CASE | computer assisted software engineering |
| CEM | common evaluation model |
| DO | departmental organization |
| E | experimentation unit of the experience-based and measure-driven architecture |
| EE | empirical evaluation sub-unit of the experience-based and measure-driven architecture |
| EBMDA | experience-based and measure-driven architecture |
| EF | experience factory |
| EIEF | empirical-investigation experience factory |
| EMAT | experimental most adequate technology |
| ESE | experimental software engineering |
| ESI | European SW engineering institute |
| GQM | goal-question-metrics method |
| I | initialization sub-unit of the experience-based and measure-driven architecture |
| IT | information technology |
| ISERN | international software engineering network |
| KEBMDA | kernel of the experience-based and measure-driven architecture |
| LP | laboratory project |
| M | measurement unit of the experience-based and measure-driven architecture |
| MM | measurement model |
| MMAT | measured most adequate technology |
| MO | ministerial organization |
| MSS | multiple (synthetic processes)-single (laboratory project)-single (pilot project) |
| OSE | object SW engineering |
| PA | public administration |
| PIE | process improvement experiment |
| PP | pilot project |
| QIP | quality improvement paradigm |
| SP | synthetic process |
| SW | software |
| TT | technology transfer |
| TTEF | technology transfer experience factory |
| UML | unified modeling language |
| VC | value constellation |
| WA | workflow automation |
| WST | weighted and scored technology tree |

serious *risk*, due to the relative immaturity of both the SW industry and SW customers, and a dangerous *inadequacy* of public research funding in information technology (IT) (PITAC, 1998). Such impact, risk, and inadequacy oblige organizations to evaluate new SW technologies carefully before accepting innovation and starting a technology transfer (TT) project. Table I of this paper presents a summary of the abbreviations used.

 

Because of the strategic importance of IT to organizations, and because of the pervasive nature of SW, the considerations noted above affect organizations of all kinds, and eventually society as a whole, rather than just the SW organizations. In fact, IT is becoming an integral part of people's lives, business, and society in general (PITAC, 1998). The emergence of the 'Information Society' is accompanied by so much transformation (Premier Ministre, 1998, 1999) that no organization can ignore IT. Doing so would mean ignoring a factor that can improve its services, expand its market, increase its contribution or profits, and eventually put its very existence at stake (Prime Minister's Office of the Hellenic Republic, 1999). Not being part of the 'Global Information Society' would be extremely risky for any country, especially those still compensating for a technology gap (Ministero, 1998); consequently, some governments are trying to establish their PA departments as the locus for enabling IT diffusion nationwide.

The present paper aims at presenting a system view of SW TT, and showing that the developed abstractions can be easily specialized and usefully applied to any kind of organization involved with SW TT. Consequently, from the stance of SW TT systems (which includes, but is not limited to, SW development organizations), this paper shows that organizations within such systems have interrelated, but specific, SW TT views and needs.

This paper is positioned on the side of the empirical paradigm. It focuses on how to help organizations to select new software technologies, understand the pros and cons of certain choices, and capitalize on related experience. It tries to say what needs to be done when an organization recognizes the exigency of a major SW innovation, but evaluates its technology awareness as not being sufficient for accepting and, *sic et simpliciter*, migrating to a new SW technology. The paper describes a process framework and a conceptual architecture for reusing TT related knowledge and experience when designing and running SW TT experiments. We abstracted a process framework and conceptual architecture from the experience we built up and the lessons we learned. We were trying to understand, from an empirical basis, which software technology was the most adequate for achieving the goals of an organization, which might be a very large, complex, and possibly very difficult to automate enterprise. The paper examines the utility of investigating empirically SW TT relative to reusing technology measurement models. The paper also examines the role that is played in an empirical investigation of TT by factors such as SW technologies, experimentation, measurement, application domain, an organization's business objectives, and by interrelationships among TT key-players and customers.

In particular, this paper is concerned with developing an infrastructure designed to answer technology-related questions on the basis of the necessary empirical evidence and, with developing know-how, reuse experiences, and eventually find ways to improve the corresponding organizational knowledge. Also, this paper makes use of and describes how to improve, contextually, SW technology models and related MMs (Cantone and Donzelli, 1998), rather than simply relying on them. However, the MMs themselves are presented elsewhere (Cantone, 1999), except for a sketch on technology models later in this paper in Sections 3 and 5, and in the appendix.

### 1.1. Related work

The background of the work reported here includes a feature-analysis-centered framework for empirically evaluating SW technologies, in the context of a SW organization (Brown and Wallnau, 1996). Such a framework is based on three sequential phases, description modeling, experiment design, and experiment evaluation.

 

- The *description modeling* phase starts from some *candidate* technologies, and aims to *locate* such technologies in the market before *selecting* the 'treatments' (Kitchenham, Pickers and Pfleeger, 1995), that is, the technologies to submit to following phases. In particular, this first phase consists of two steps, one that investigates the *technology domain genealogy* and the other that maps that genealogy onto the *problem domain*.
- The *experiment design* phase tries to generate both *experiments* and *evaluation criteria* starting from the located technology and the *business objectives*. It first analyzes fully and comparatively the *features* of the located technology, and then follows with two more steps. One step is aimed at formulating the *evaluation hypotheses*; the other is aimed at *designing experiments* that can generate evidence about the benefits of the technologies with respect to the business objectives.
- The *experiment evaluation* phase analyzes the *business domain* in depth. It investigages the *compatibility* of the technology with the *internal situation* of the organization, implements *demonstrator studies*, develops *benchmarks* and other laboratory techniques, and eventually gives answers to technology related questions.

*Experimentation* is central to the present paper. The milestones of SW engineering experimentation started being placed from the middle 1980s (Basili, 1992, 1996; Basili, Selby and Hutchens, 1986). They made a research paradigm shift to an *experimental SW engineering* approach. On the basis of such milestones, on the conviction that SW-engineering experiments need to run across *multiple environments*, and on evidence that software experimentation depends on *multiple influential factors*, the ISERN was established in the early 1990s, and ESE groups grew (Rombach, Basili and Selby, 1992).

Starting from the middle 1990s, efforts were made to reach practitioners. Research results were diverse and included the following.

- Experimental basis of SW engineering (Burgess, 1995).
- Leading guidelines for organizing experimental studies for software method and tool evaluation, so as to obtain meaningful results (Kitchenham, Pickers and Pfleeger, 1995).
- Main concepts, processes, and constraints for running formal software experiments (Pfleeger, 1994, 1995 a–d).
- A basic method for evaluating SW-engineering methods (Kitchenham, 1996a–c, 1997).
- A distinction between the quantitative and qualitative evaluation methods for SW tools and methods (Kitchenham, 1996a–c, 1997).
- A classification of the evaluation methods for SW tools and methods, including their pros and cons, the related influential factors and technical selection criteria, and their risk (Kitchenham, 1996a).
- The principles of SW feature analysis (Kitchenham, 1996a–c, 1997).
- A method for developing SW quantitative case studies (Kitchenham, 1996a).

Moreover, many papers report lessons learned while running new or replicated experiments (Basili *et al.*, 1996; Briand *et al.*, 1996; Ciolkowsky *et al.*, 1997; Kamstie and Lott, 1995; Porter *et al.*, 1995; Visaggio, 1997; and see the ISERN web page for further references (ISEN, 1999)). Finally, a PIE project was launched in Europe, to develop a standard-like TT meta-model (see the ESI web page (ESI, 1999)).

Our work reflects our adopting the following positions. SW is an experimental science. It evolves by cyclical quality improvement processes (QIP (Basili, 1993)), where improvement iterations are also

 

learning cycles (Garvin, 1993). In order to have learning capitalized for reuse, SW companies can install an EF (Basili, 1989; Basili, Caldiera and Rombach, 1994a). This is a conceptual architecture that is adequate to enact QIPs. It needs to be specialized to the particular organization and application domain (e.g. component factory (Basili, Caldiera and Cantone, 1992)). SW can be influenced by many factors: technical, social, and economic. In order to detect significant factors we can apply a goal-driven approach. This usually helps to break down the SW problem. It involves modeling the entity of interest in terms of the influential attributes, measuring such attributes, and eventually developing a MM for the overall entity, for example by using GQM (Basili, Caldiera and Rombach, 1994b; Basili and Rombach, 1988; Van Solingen and Berghout, 1999). Experimentation is necessary to verify modeling hypotheses (Zelkowitz, 1994). It has to be kept under control and involves activities that call for coordination. The validity and exportability of the results depend on the characteristics (or parameters) of the experiment (Zelkowitz and Wallace, 1998): for example the environment (field, laboratory, or other), enactors (novices, experts, or a mix), product type (synthetic processes, pilot project, or full projects), product cardinality (single or multiple), and results presentation (descriptive, correlative or cause–effect).

The remainder of this paper imports, reuses, abstracts from, extends, and summarizes knowledge that is drawn from many related works, without citing all of them.

### 1.2. Contribution and novelity of this paper

With respect to related and prior works, this paper can be characterized as follows.

- This paper emphasizes the role of *measurements* and *reuse* of experience in the empirical investigation of SW TT. In particular, it begins by specializing the concept of EF to both SW empirical investigations and SW technology modeling. Then, it merges them, utilizing the reuse of SW TT models in the selection of technologies to perform empirical investigation, and utilizing empirical results to accredit or to discredit the technology models.
- This paper takes into consideration the *system* of organizations that are involved with SW TT, rather than that of SW developers only, and examines the roles and the *diverse* interests, possibly conflicting, among the various actors in the TT network. This can include complex organizations that are the *non-SW organization* ‘end–customers’ of SW technology. Sometimes, when these are large and important actors in the TT system, they may affect the choice of the technology to employ in SW construction and modification.
- While a described TT system concept can be applied to an organization of any kind (‘reference organization’), this paper, which is based on field experience, assumes the *point of view* of ‘innovation-developers’, rather than as re-engineers, maintainers, or ‘developers’, as is more usual in the SW literature.
- This paper also considers TT as having major effects on the work-processes of the reference organization (*TT in the large*).
- While TT evaluation models known from the literature often relate generically to goals and organizational contexts, this paper specializes abstractions in order to develop a detailed architecture and process model suitable to empirically support TT investigations for specific, realistic, and *complex classes* of organizations, goals, and SW technologies.
- The paper shows how to proceed from abstractions to real work. It focuses on a specific SW technology, the WA SW maintenance and development technology, in the context of a non-

 

rigorous approach to BPR (Berztiss, 1996; Hammer and Champy, 1995), and in the context of a PA function (AIPA, 1997). We use the expression 'BPR' in a broad way to refer to any kind of effort aimed at changing the business processes of an organization, ranging from radical changes (Hammer and Champy, 1995) to more continual and evolutionary changes, including WA.

- Finally as a short case study, the paper describes a specific reference organization of the Public Sector, where WA is a new technology. A *process model* for running empirical TT investigations is derived. It is an affordable model, because it is partially, rather than comprehensively, based on multiple experimental developments. It can be further simplified, when tailored to cases that are less complex, or when empirical-investigation contexts are more mature than that reported here.

### 1.3. Paper organization

The remainder of this paper is organized as follows. Section 2 presents the SW VC and the SW TT system. Section 3 recalls the CEM, and discusses its utility for experimentation. Section 4 describes the process framework and conceptual architecture that this paper has adopted and enhanced, to develop empirical TT investigations. Section 5 summarizes our TT experiences and presents a brief case study. The conclusions in Section 6 look forward towards future work. The appendix shows an application of the CEM to evaluating the utility of an empirical evaluation.

## 2. VALUE CONSTELLATIONS AND TECHNOLOGY TRANSFER SYSTEMS

SW TT consists of introducing or switching to a different SW technology. Sometimes, in the SW research community, SW TT is intentionally limited to SW maintenance and development organizations (the so-called 'SW sector'). In our conviction, because of the characteristics and the specific pervasive nature of SW technologies, there is no reason or utility in applying *a priori* limitations to the generality of SW technology models and to the related experimentation and measurement models. *Vice versa*, extending SW TT models and methods to encompass the needs of all the possible SW TT organizations can help the SW sector in improving its business. Thus, this paper also emphasizes the point of view of non-SW organizations. Due to such an approach, in the remainder of the paper, some organizational concepts are used. These are sketched out in this section, so that qualified readers can skip over them without suffering technical losses.

A VC (Norman and Ramirez, 1993) is a system of organizations which depend on each other to co-produce value at least as fast as their competitors do, and to possibly behave and perform as well as their customers would want. Mutual-help relationships rather than hierarchical relationships are established among VC members. A VC is established by an explicit act of the members, and is based on mutuality, coordination, and specialization. A VC is not a partition. In fact, coherently with the concept of specialization, some organizations may be member of multiple, even competing, VCs. A VC has end-customers, which are the destinations of its final products or services. A VC includes a leading organization, or 'client-leader', which manages relationships with, and is recognized in such a role by, the end-customers, whose needs eventually lead the strategic choices. Other types of organizations that participate in a VC are developers, and end-users. In order to reach its goals, a VC enacts TT. This paper emphasizes VCs whose leadership is, or should be, played by a PA.

 

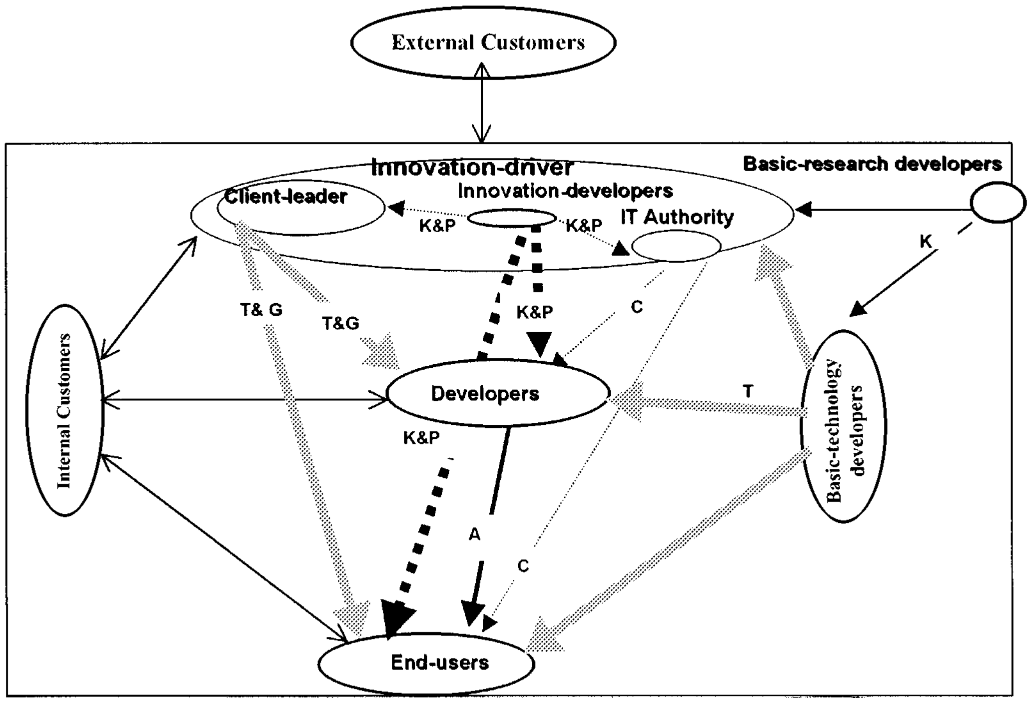


Figure 1. VC and TT system. The abbreviations used in the figure are: K, knowledge; T, technologies; G, guidelines; P, prototypes; A, applications; C, control and regulation.

We are mainly interested in the *SW TT System*. This is a network of TT key-players, i.e. organizations that significantly influence TT processes. Key-players perform specific roles and are strongly related to each other. The results of TT processes depend both on the roles and the associations among key-players. Furthermore, depending on the dynamic content of the relationships, the behavior of the key-players may change over time.

TT is essential for any kind of organization, VCs included. Any TT usually directly involves two or more organizations, and may indirectly involve other organizations, as shown in Figure 1. For the sake of simplicity, *Suppliers*, i.e. organizations that perform the role of interface among the other actors in the TT system, are not further considered in this paper. Instead, emphasis is given to the needs for evaluation methods of various other types of organizations. In the SW TT system these include the following.

- *Basic-technology developers* are necessary. They introduce new or innovated technologies, development methods and tools. Before starting a new project, these organizations need to predict the value that the technology is expected to produce for the target customers.

 

- *Basic-research developers* have to be presumed. They develop research results that could, in the long term, be included by new technologies. An example is universities and other research centers that are instances of basic-knowledge developers.
- An *Innovation-driver* is necessary. It pioneers the experimentation and dissemination of new technologies. Technology evaluation is an essential task, in some cases it is the mission of this type of organization. Innovation-drivers fall into the following categories.
    - ◆ *Innovation-developers*, usually small business or applied research institutions, which implement advanced processes, and use advanced technology and knowledge, to develop new knowledge and prototypes that are usable by other organizations in the TT system, in the present or future market. For example, by the definition above, academic ESE groups, and enabling technology centers (PITAC, 1998), that experiment with technologies with the intention of generating up-to-date technology knowledge to transfer to some other organizations, are innovation-developers.
    - ◆ The *client-leader* organization is the organization that leads a VC. A client-leader uses its power to help the overall system (i.e. all of the VC members) to evolve. It receives mutual help from the remaining VC members. A client-leader influences the choice of new technologies, introduces and tests them in its own processes, and delivers assessed technology and guidelines to members of the VC, as expressions of its leadership. The client-leader organization focuses TT processes on needs of the VC's end-customers. Relevant exceptions to these behaviors can be observed in the public sector, where the establishment of a proper PA-centered VC is effectively the responsibility of governmental organizations. These can include an *IT Authority* for indicating, advising, regulating, or controlling PA-wide advances in IT technology and application to business processes: see, for instance, (PITAC, 1998) and (AIPA, 1997).
- *Maintainers and developers*. SW maintenance and development (SW sector) organizations are also necessary. They play the role of the direct receiver of the technology innovation. These organizations apply SW technologies to develop their products and services, and may try to employ the technologies that are the most adequate to their business goals and application domains. For such reasons, they need to evaluate new technologies with respect to the goals, baselines, and contexts of their own organizations. In order to improve their technology, they may also want to acquire and use the knowledge developed by innovation-developers. A common goal of maintainers and developers is to satisfy the needs of their customers (end-user organizations). Consequently, SW sector organizations also aim at having the technologies that are the most adequate to maintain and develop products that meet the needs and business goals of their customers. Moreover, when part of a VC, maintainers and developers can be requested by their client-leader organization to specialize and transfer technology to associated end-users. Finally, maintainers and developers can participate in more than one VC. Hence, because of their aim, organizational duty, and market position, maintainers and developers may also want to evaluate new technologies from the end-user point of view.
- *End-users* represent the final destination of technology improvement. They adopt the products built by the organizations listed above for delivering goods or services to their own customers. Again, they try to acquire and use the applications most suitable for their needs and goals. Instances of the common goals of *customer-service organizations* are to improve the quality

 

and performance of both the relationships with their customers, and the workplace. When part of a VC, end-user organizations can receive tested and assessed technologies from the VC's client-leader and the maintainers and developers. End-users are expected to give further accreditation to such technologies in the context of their organizations. They may want to understand which technology is, from their own point of view, the most adequate for their own systems, either directly or by using the knowledge developed by innovation-developers. In order to prevent misunderstanding, let us explicitly note that 'the most suitable technology' to satisfy some stated needs, is not the same as 'the best technology'. The latter is a superlative expression that is not reasonably applicable to advancing technologies for the development of innovative custom SW.

An end-user organization may also want to keep its own *service technology* (i.e. tailored SW, which supports the execution and control of processes for delivering its products and services to its customers) under the control of its own management. This is because of the strategic importance of IT and the contextual immaturity of the actors in the global information society, SW industry included, noted in Section 1. In fact, for a leading end-user, keeping the knowledge, which is related to the evolution of product or service technologies, out of the organization's control could transform a customer–maintainer relationship into a strategic dependence of the former on the latter. This assumes the absence of an end-user-centered VC that includes developers and maintainers. Keeping technology under management control includes knowing, in depth, what applications do, how they behave and affect products and services, and what are their pros and cons. It should also include knowing the applications' basic architecture, distinguishing general from customized parts, and why, where, and how maintenance interventions have introduced changes. With respect to such aspects, an experimental hypothesis follows, which is assumed by our TT research work.

**Hypothesis**. *The closer the SW design, construction, and maintenance abstractions are to the domain abstractions (possibly, they are the same abstractions that end-users use in their everyday work), the more the end-user organizations can keep the SW applications under their control.*

Consider an example. In PA, a MO, which uses WA applications to manage and enact its business processes, is an end-user organization of the WA technology. Based on its goals, the MO, as the of buyer of customized WA applications, may want to negotiate the technology to be used for developing and maintaining the same applications. Based on past experience, the MO could also want a technology, whose evolution can be controlled as easily as possible by the MO management. Maintenance or development technology, or both, could be thus requested to incorporate a visual, end-user, domain-oriented language with primitives for immediate definition, direct operation, and integrated management of *organization-charts*, *people* and *groups*, *roles*, *documents*, *message* and *document passing*, *competitive tenders*, *work-processes*, and so on.

- *Customers*. All the organizations above have internal customers (managers, employees, and sub-organizations), and external customers. *Internal customers* can negotiate and eventually affect TT. Internal customers' unions are expected to be able to evaluate the effects, pros and cons, of TT from their point of view. *External customers*, in turn, can be organizations (e.g. firms) or people (e.g. citizens). *End-customers* of a VC are the external customers of the customer-leader organization (those of a single company, i.e. a company that is not part of a greater organization,

 

are the company's external customers). In fact, in a compound organization, for example, a VC, a customer of an intermediate organization can be a supplier. While external customers cannot directly affect TT, the needs of the end-customers are the main drivers of management initiatives, TT initiatives included. For example, in a PA, the external customers of a DO, whose mission is to supply agencies of the common super-organization (e.g. a MO) with furniture or maintenance services, are also suppliers. Fundamentally, the needs of the super-organization's external customers, eventually citizens and firms, should drive the DO management.

All of the organizations of the SW TT system need to evaluate technologies. Innovation-developers are expected to be able to develop all the necessary models. In the following, the paper assumes the point of view of such an innovation-developer and calls the *reference organization* any organization that is customer of, or is assumed to be a potential customer of an innovation-developer. A reference organization can be any of the TT system's organizations above. Obviously, the TT knowledge produced by innovation-developers depends on the goals and other characteristics of the specific reference organization. In fact, with respect to advancing SW domains, technologies can change more quickly than the ability of some organizations to learn. Hence, it could be impossible to find, at a certain point in time, a technology which is the most adequate to satisfy the needs of the multiple actors of the TT system. For example, due to learning time and costs, a developer could evaluate as 'not convenient to use as a maintenance or development technology', the one that the customer evaluated to be the most adequate to meet the goal of keeping a new SW application under the control of its own management people. In other words, a certain technology, for instance a WA development system, could give perfect results for a certain end-user, interesting results to some other end-users, more or less adequate results to some developers, and completely inadequate results to some other developers.

In the following, we use the term organization to denote the reference organization, unless otherwise indicated.

## 3. MODELS AND EXPERIMENTATION: AN INEXTRICABLE DICHOTOMY

One of the most frequently asked questions in the every-day life of managers in organizations is whether to rely on models of the reality, or run tests or experiments before making a choice. Underlying the questions is a familiar dichotomy.

- While individual or organizational experience and knowledge are never given or acquired for free, the experience and knowledge available at a certain point in time, when conveniently organized, can be reused, with little additional cost and time, to confirm or reject hypotheses. Conversely, running experiments consumes time and employs scarce resources.
- In order to simplify understanding, and to consolidate, structure, manage, extend, and transfer knowledge, models are conceived. These models abstract from reality and experience that is in turn part of the reality. In the best case, an interpretation of a model maps the reality correctly, but partially. In addition, models can be wrong. They can miss facts that should not be neglected, and eventually they become obsolete—no longer adequate to represent new, relevant perceptions of the reality. Conversely, the continual, rough use of case-by-case experience and the systematic rejection of modeling are antagonistic to science and engineering, are opposite to the interests and culture of a mature industry, and are contrary to human rationality.

 

In essence, a MM is also an abstraction from experience, and a mapping from an empirical relational system to a formal relational system. Conversely, in order to get empirically new higher-level knowledge, any experiment makes use of modelled lower-level knowledge.

In conclusion, in order to evaluate an entity, a tradeoff exists between the application of the available models and the development of new experiments. To make a choice, in other words, means to decide whether the accumulated experience is convenient enough, or whether the new experience is feasible, affordable, and necessary.

Even though it ignores the question of when to acquire a new technology, a means for answering the above question is to evaluate the utility of the experimentation. Utility depends on both the characteristics of the experiments, and those of the organizations. For such an evaluation, the goal-driven CEM (Cantone, 1999) can be applied. The appendix shows an example of such an application.

## 4. PERFORMING SW TT EMPIRICAL EVALUATIONS

In order to investigate SW TT, we need to draw upon concepts, models and practices from diverse research areas, such as process re-engineering, empirical evaluation, goal-based improvement, domain-based improvement, continual improvement, model and measure development, and benefit versus cost evaluation.

Moreover, when organizations are concerned with SW TT, two notoriously difficult problems merge, increasing the complexity of the overall problem. They are how to develop and maintain, and how to transfer, value-adding SW technology. Both represent immature knowledge fields. Up to now, SW and TT experiences have not yet been abstracted enough to hold independently from specific organizational factors (e.g. baseline, goals, people, application domains and customers, etc). Also, specialization criteria have not been yet been refined enough to be easily applied to specific organizational factors.

Consequentially for these considerations, while preliminary experimentation and empirical evaluations are welcome practices as a part of any TT, they should be strictly enforced for SW TT, whenever the utility of the experiment (see the appendix) goes above a minimal threshold. Finally, let us note that, while there should be no development/production process without measures, obviously there is no valid experimentation/empirical-evaluation process without measures and *evaluation criteria*.

### 4.1. Cost-evaluations versus benefit-evaluations and feature-evaluations

Experimentation with TT cost-evaluation introduces problems that usually do not occur, or that we can easily transpose or conveniently neglect when coping with technical features or benefits. In fact, some costs, for instance some risk costs, present subjective rather than objective determinations because their character and magnitude depend on the perceptions of the organization rather than relying on prices (such as evidenced by free markets).

Moreover, it may be difficult to think about experimental laboratory processes for TT cost evaluation. In fact, we are unable to isolate and predict some of the costs involved with TT. Some, because of their nature, can be only observed in the field. Instances of such costs are the technology’s social costs, costs of the impact on the organization, and costs of adaptation to the ‘informal organization’ and its relationships.

### 4.2. Conceptual architecture

Figure 2 shows an EBMDA for running empirical TT investigation processes. The goal of such a process is to become aware of technologies generically available in the market, detect those items of the total technology set that more than others meet the (reference) *organization goals*, and eventually from them an organization-wide accredited technology item that is used across the organization.

The *business objectives* of the organization drive the execution of the experimentation processes. Secondary drivers are the *application domain* and a characterization of the organization's *customers*, for example the *customer goals*, their *organizational needs*, their *competencies*, and their position in the market (*segment*). The kernel of the EBMDA (KEBMDA) includes parts from both a TTEF, which is aimed to learn about TT, and an EIEF, which is aimed to learn about experimentation. In this approach, the KEBMDA includes the following units:

- M, mainly a TT measurement modeling unit, and
- E, a processing unit to develop empirical evaluations.

With respect to the position of Brown and Wallnau (1996), beside the unit E, the EBMDA makes the need for applying the EF concept and using measurements systematically explicit.

The M supplies and handles *evaluation models*. In addition to providing *empirical–evaluation utility models*, it produces *feature models*, *benefit models*, *cost models*, and *value models*. M is part of the TTEF. With respect to a previous specialization of the EF model (Basili, Caldiera and Cantone, 1992), the TTEF deals with technology models rather than with SW components. M supplies E with technology experience and knowledge, and with continually improved TT models. M asynchronously develops *models* by itself. It exchanges models with other organizations, both internal organizations (*knowledge exchange*), and external (*imported knowledge*). M stores MMs and other related knowledge into the MM-Base.

The E (see Figure 2) is intended to perform empirical investigation processes. E is part of the EIEF. The experimentation knowledge base feeds E with *empirical investigation models* (e.g. the models described by (Kitchenham, 1996a) or those presented in sub-Section 4.6) and with related *evaluation criteria*.

Models need to be continually accredited and improved. While the TTEF and EIEF instruct and supply E, both learn from the successes and failures of E. This is obtained through the *feedback* provided by E and, after the release of technology, by the implementation processes. It is evident that, at a certain stage in the evolution of a learning organization, technology domains become relatively well known, and the TT models become relatively stable and reliable. Consequently, relevant empirical processes can be strongly simplified, and applied less frequently. An association is hence established between the TTEF and EIEF, which eventually lays a basis for additional abstraction of the EF concept.

In this approach, E is structured as follows:

- I, to initiate the empirical processes to be run (see sub-Section 4.4), and
- EE, to do empirical evaluations (see sub-Section 4.5).

### 4.3. Measurement

M supplies TT investigations with technology models and MMs. Figure 3 aims to give a high-level view of the technology models that M could supply to E.

 

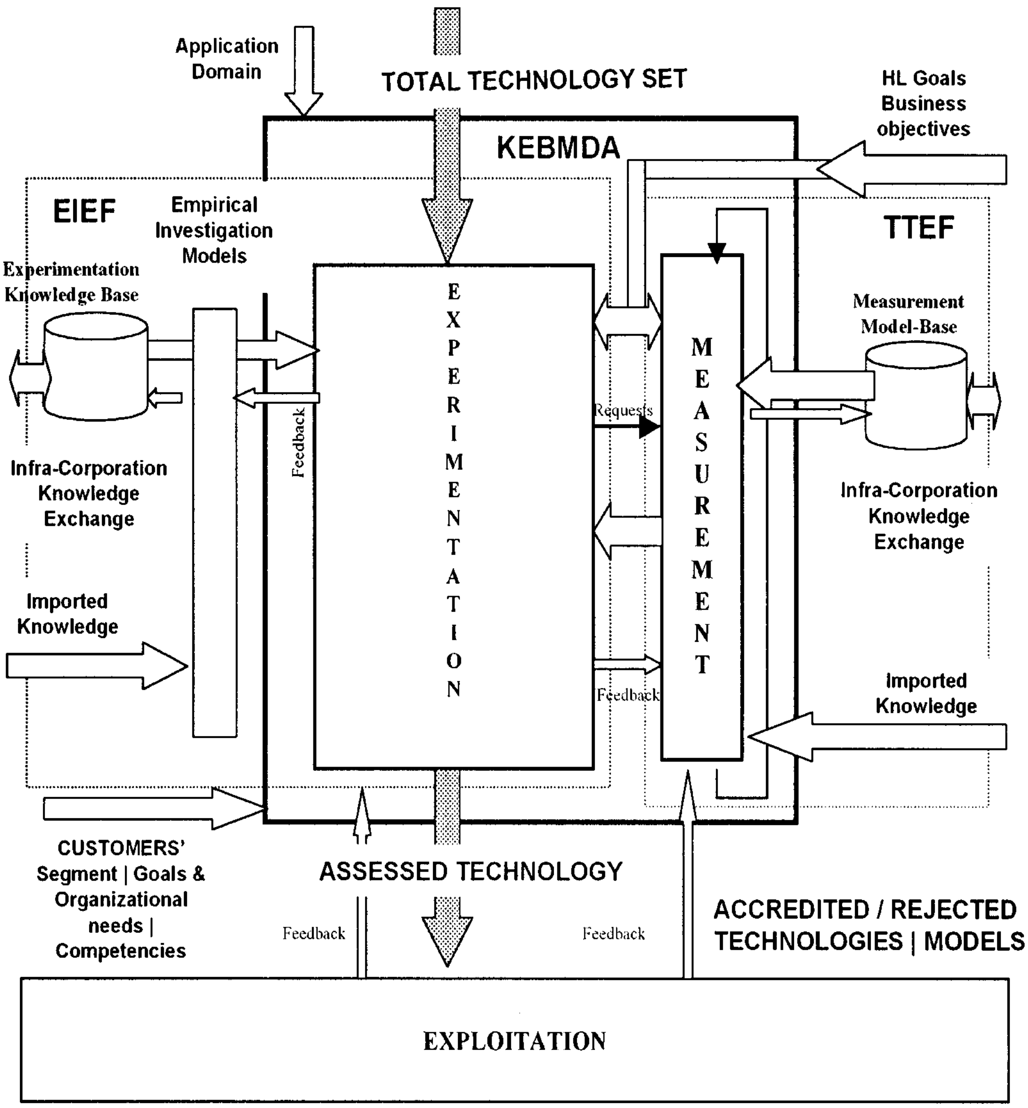


Figure 2. The architecture for enacting measure-driven empirical evaluations.

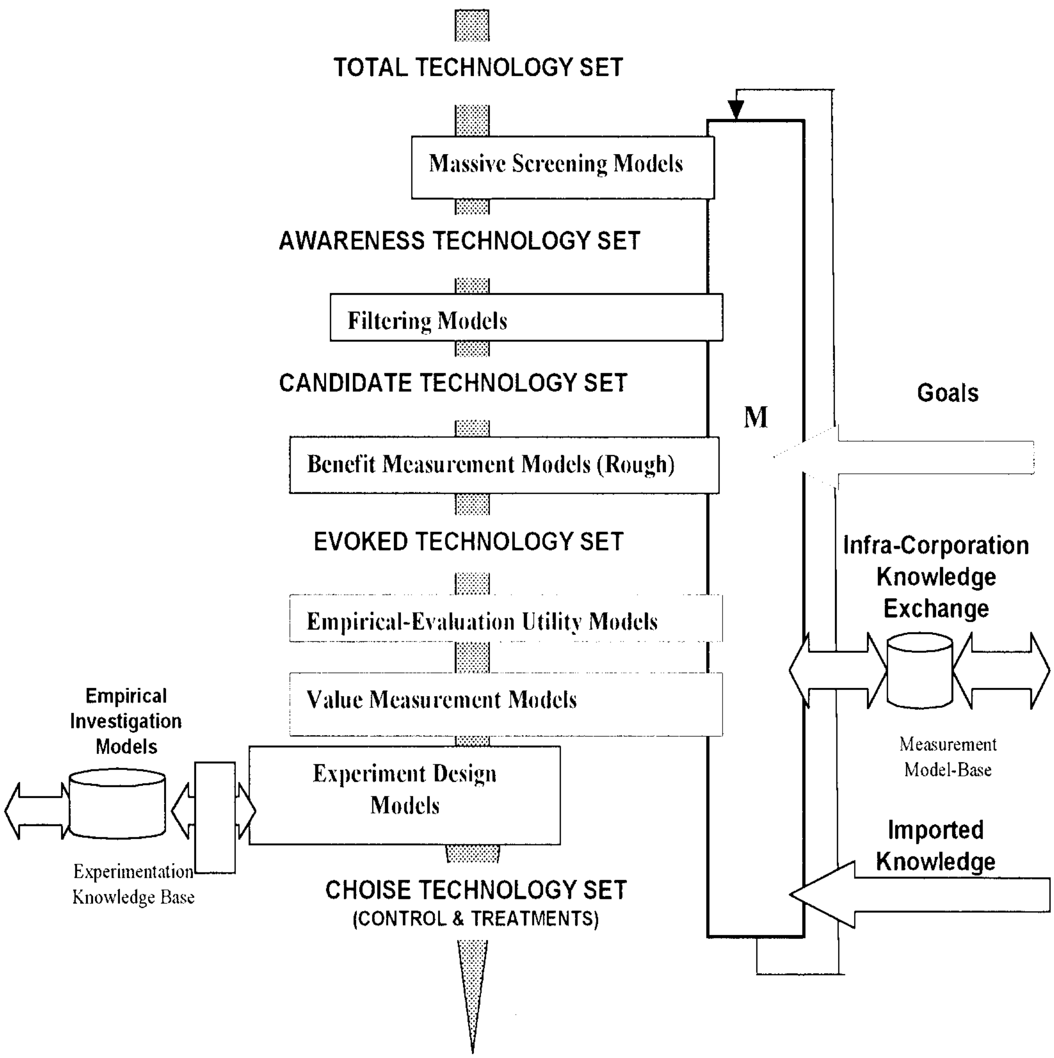


Figure 3. Using MMs to work on technology.

Model instances that can help in developing TT evaluations are:

- *on/off models*, to screen the technology currently present in the market and produce the *awareness technology set*. Such a model usually works on the technologies' general documentation, taking into account attributes such as the presence of the producers in the market and the technologies' technical features;
- *feature models*, to screen awareness technologies, and produce the *candidate technology set*. These models usually deal with the technical features, with the aim of expressing the desired/expected/actual/prospective features of a technology;
- *benefits MMs*, to express the desired/expected/actual/prospective benefits of a technology;
- *cost MMs*, to express the desired/expected/actual/prospective costs of a technology; and
- *benefit versus cost MMs* (or *value models*), ratios, differences, and trends.

Starting from the candidate technology set, the MMs attempt to extract, step-by-step, the more manageable technology subsets. In order to develop such MMs, M performs a goal-oriented measurement life cycle that, starting from the high-level organizational goals, allows the management to identify what kind of information should be available to achieve those goals (Cantone and Donzelli, 1998). By such an approach, M becomes an integral component of the management activities used to generate, refine, and achieve the organizational goals, whereby the final MMs become items of a purpose-fitting measurement plan (Cantone and Donzelli, 1999).

In order to continually improve HHs, M as part of the TTEF learns from the successful and unsuccessful applications of the models. Based on the information feedback, M proceeds to maintain models—that is, to correct, enhance, and abstract from existing models, and adapt them to new needs. In particular, M abstracts from the multitude of models it handles, and attempts to develop unified models, thus reducing the complexity of the system of models to be managed. Finally, on request, M actualizes instances of the model abstractions, to deliver models that are tailored, as much as possible, to the requestor's needs. Hence, M behaves as a model developer and an instance enactor rather than only as a simple server. Instances are specialized to specific requirements (e.g. goals, application domain, constraints, people, benefits, costs, etc.) and fitted to the precision and granularity needed by the specific destination (e.g. construction and the evoked set or the chose set, respectively).

### 4.4. Initialization

The I encompasses activities that are preliminary to the development of the core part of the TT empirical investigations. It includes the preparatory work necessary for evaluating the opportunity for, and eventually running of, the experimental process (see Figure 4).

In general, TT is part of what can be an intensive BPR effort. The initiator is activated during the BPR life cycle, usually after both the analysis of the as-is system and the design of the changes (AIPA, 1997) have been performed. At that point, (possibly alternative) solutions will have already been identified, and technologies have been roughly identified as opportunities to reach the goal (Harmon and Watson, 1997; AIPA, 1997).

Initiating the experimentation requires the following activities:

(i) macro analysis of the experiment feasibility, and a rough evaluation of the experiment's execution time and upper cost bounds;

 

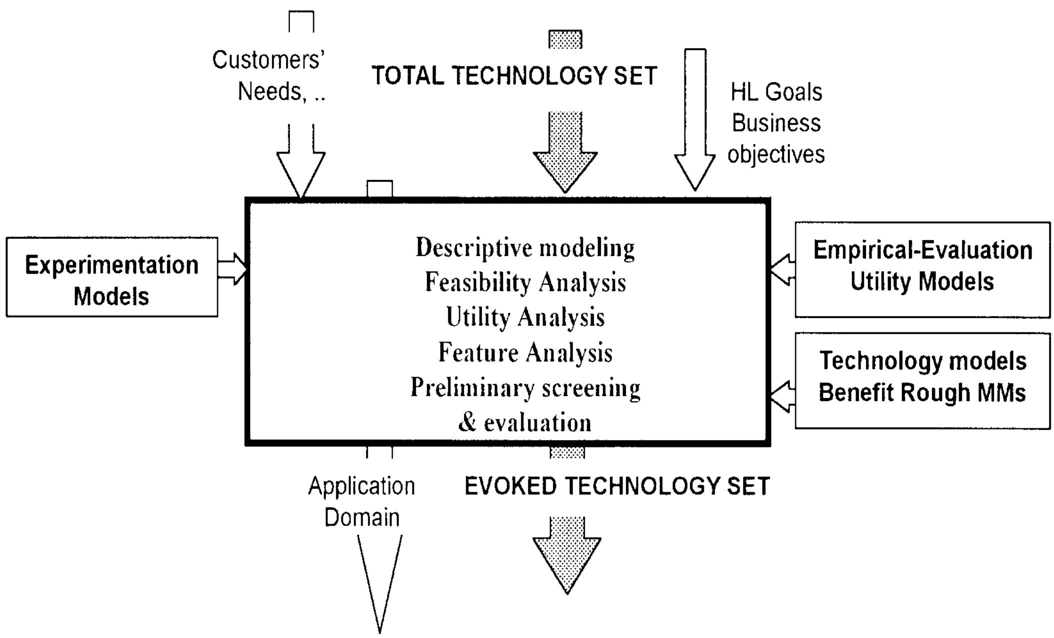


Figure 4. Initiating a TT process.

(ii) acceptance of the experimentation by, and commitment to the experimentation by the management of the organizations involved;
(iii) the formal definition of the reference organization's goals and constraints;
(iv) the development of a preliminary analysis of the application domain (e.g. the basic analysis of business system processes, both as-is and in the TT view), and characterization of the customers;
(v) the detection of the SW technology actually in use at the reference organization ('control' (Kitchenham, Pickers and Pfleeger, 1995));
(vi) the identification of alternative BPR strategies;
(vii) for each alternative strategy:

  (a) the identification of alternative types of intervention, e.g. light process re-engineering, radical process re-engineering, WA, CASE tool change, etc.,
  (b) for each alternative type of intervention:

    - the definition of the involved technology domain, for example WA technologies,
    - the preliminary screening of the technology set of the particular domain, for example WA application development systems, by using on/off models,
    - the determination of the candidate technology set, based on M's guidelines and feature-based descriptive models,

 

- the preliminary identification of the experimental model to adopt and relate evaluation criteria,
- the refinement of the feasibility analysis, including a re-evaluation of the experiment's expected development time and utility,
- the decision to proceed, change, suspend or cancel the alternative TT evaluation process.

Following the preliminary activities above, the initiator:

(i) describes the experimental models to adopt and
(ii) extracts the evoked technology set from the candidate technology set with the guidance of benefit MMs.

### 4.5. Empirical evaluation

The EE performs the necessary TT investigations. It aims to supply the organization with useful technologies, i.e. technologies that are able to meet the organizational goals. During the construction of the empirical evaluation, the experiment is designed and performed (see Figure 5).

In this approach, in order to improve the value of the technology selection process, the experimental design and construction are based on collaboration between two tailoring activities. One activity is aimed at instantiating reliable MMs that are able to minimize the number of competing technologies (zero and one would annual the need to proceeding further to an experimental process). The other activity is aimed at choosing and designing the experimental process most appropriate for, and tailoring it to, the particular case.

While on/off models were used for becoming aware of, and locating technologies, and while benefit models were informally employed for evoking technologies, EE necessitates the use of finer, value-oriented MMs.

During the design, an *experimental technique* is chosen. The choice strongly affects the experimental process (see the next sub-section). During the design, the *experimental subjects* are also located and assigned. The experiment is organized, and the schedule (the roles and the number of people involved) is planned.

The *experiment objects* consist of a *control technology* and one or more *treatment technologies*. The former is the SW technology presently in use. The latter technology set is made up of those items of the choice technology set, which reported the best measures, i.e. the MMATs. One or more items may be chosen depending on the resources assigned to implement the experiment.

### 4.6. An affordable process for evaluating new technologies

SPs are conceptually autonomous and separable parts of a project, which, in the opinion of experts, are able to represent the complexity of the project in full (Zelkowitz and Wallace, 1998). With respect to BPR applications, SPs are able to characterize the business processes of the overall reference organization, in the opinion of the experiment designer.

A maintenance or development group, termed the *synthetic group*, is a group in the organization that, in the opinion of experts, is representative of the organization's average ability to perform projects in a certain application domain.

 

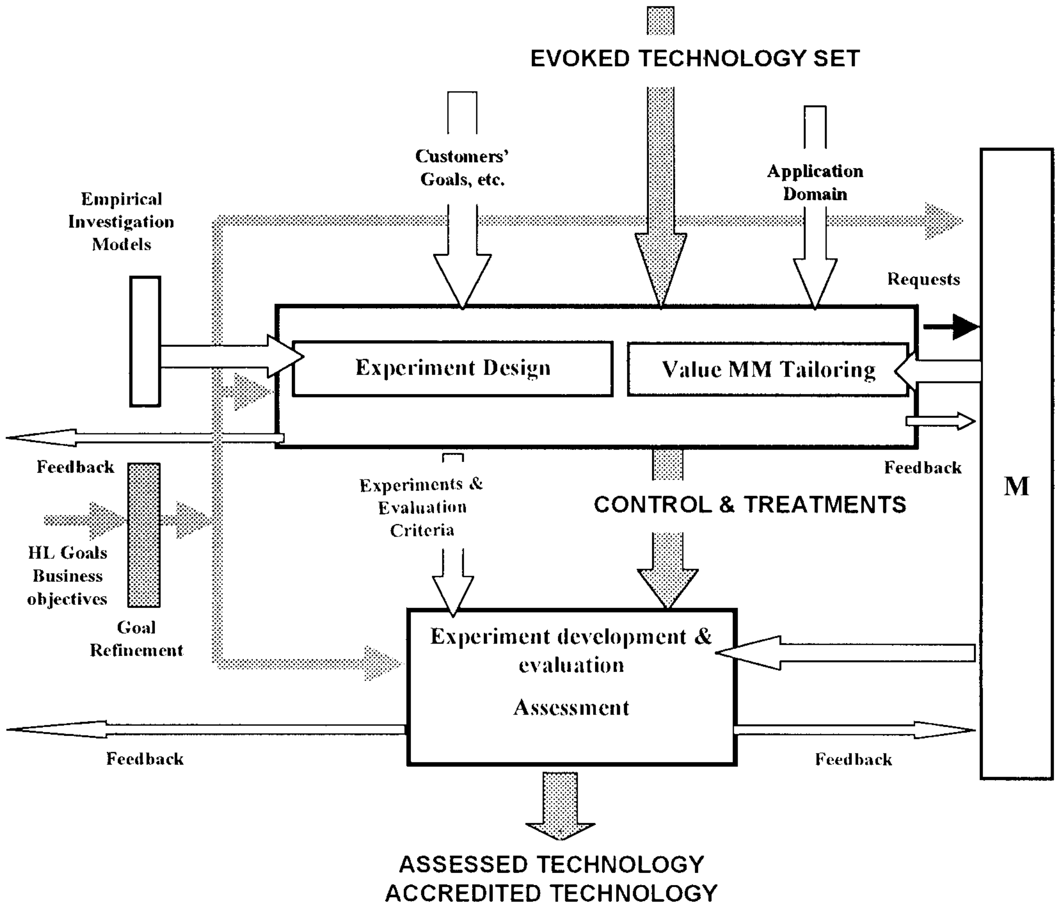


Figure 5. Enacting experiments and evaluating results.

Let us consider a technology evaluation technique that is based on the following steps:

(i) multiple developments of SPs;
(ii) a single development of a LP, which is a scaled-down realistic case study that is less than a pilot project but more than a toy project; and
(iii) a single development of a PP that is an observed field project.

We denote such a technique by MSS, as in multiple–single–single. By MSS, the construction of an experiment includes the following design activities:

(i) to identify the PP. For example to identify an end-user organization of the PA, e.g. a division, within which to introduce WA;

(ii) to identify the SPs that are able to characterize the overall business system and possibly, for the sake of reuse, the application domain (i.e. an extended class of homogeneous business systems). For example to identify characteristic administrative workflow processes of the PA;
(iii) Scale down the PP and extract the LP. For example, referring to the PP instance above, to identify the kernel workflow processes of the division, scale down the number of platforms, simplify the user interfaces, and so on, up to define requirements of a WA basic application for the division, generated from the laboratory development.

The construction of MSS experiments also includes the following implementation activities:

(i) develop the SPs. When these are compatible with the available resources, arrange the SP developments as formal experiments (Pfleeger, 1994, 1995a–d) by using, in addition to the control technology, one or more treatments (multiple developments). Otherwise, use a synthetic group, employ one treatment, i.e. the MMAT, and manage the group allocation on a couple of developments, in strict time-sharing (virtual parallelism);
(ii) develop the LP by using the EMAT, as determined by applying evaluation criteria to SP developments, and a synthetic group;
(iii) develop the PP by using the most adequate treatment again (as confirmed by the LP development), and a synthetic group.

All of the experimental developments described above should be successfully completed, before releasing a new technology organization-wide.

The MSS technique is relatively expensive and time consuming. In fact, MSS is aimed at evaluating technologies that are completely new for the reference organization and, in such a context, for the innovation driver. Usually, this is the case of PA with respect to WA technologies.

Moreover, MSS is more affordable, in terms of cost and time, and easier to manage than running formal experiments in full. This is because MSS is based on the concept of the multiple development of *synthetic processes*, and it can use the concept of a *synthetic development group*, rather than the concept of parallel developments of a PP by randomly-arranged development groups. In addition, the MSS LP work is mandated to produce deliverables, which are almost completely reusable, by the PP. Consequently, while the LP additional development process slows down the overall empirical process, the costs of LP come partially from the PP costs, while its products almost completely go to compose the PP products.

The results from the MSS technique are less reliable and less portable than results from formal experiments. In fact, while MSS developments can include virtual parallelism, formal experiment developments evolve in real parallelism, and include both physical and logical redundancies. In the MSS technique, hazards placed by synthetic processes can be strongly mitigated by the successive development of the LP and PP. On the other hand, the hazards arising from the use of synthetic development groups may be unavoidable. The limits in the criteria applied to select technologies, may be too constraining. Consequently, MSS processes should be very carefully evaluated, when considering whether or not to export/import their results.

A summary of the EE process follows. As already stated, such a process was put in place for evaluating new technologies, for a BPR context showing scarce familiarity with experimental processes, and for situations where stable measurement models are not available.

 

(i) Obtain utility, benefit and cost TT MMs and guidelines from M that are specialized to the specific goals and (alternative) BPR strategy.
(ii) Select the technology treatments, by applying some of the TT MMs sketched in the previous sub-Section 4.3, to have the MMAT set.
(iii) Let the MSS technique be the empirical investigation technique to apply.
(iv) Design the experiment objects:

  (a) design SPs,
  (b) analyze and design PP,
  (c) extract LP from PP; adapt and design LP.

(v) Identify the experimental subjects.
(vi) Define the evaluation criteria.
(vii) Construct the experiment, collect data, and evaluate the results empirically:

  (a) construct multiple SPs,
  (b) construct the LP,
  (c) construct the PP.

(viii) Give the M results for MMs accreditation and TT assessment. Give feedback information for improving the MSS investigation model.
(ix) Release the TT for implementation organization-wide.

The above process should be easily tailored to other investigation contexts. For instance, to give less emphasis to the LP development, where experimental maturity is available and SPs can be conveniently arranged. Or, for instance, to give more emphasis to the measurements where a stable and reliable set of technology models is available (see also the appendix).

## 5. APPLYING THE MEASURE-DRIVEN PROCESS FRAMEWORK: A CASE STUDY

We applied the architecture and process above in a real project, where we performed the role of the BPR innovation-driver. We carried out the laboratory work in the Laboratory for Software Technology Transfer at the Administration and Documentation Computing Center of the University of Rome at Tor Vergata.

Based on the contents of the previous sections, we performed the following activities:

(i) Identified a reference organization and developed an understanding of its context and goals.
(ii) Evaluated the utility of the empirical evaluation for the reference organization.
(iii) Identified a PP in the reference organization.
(iv) Derived BPR interventions and the related technologies.
(v) Modeled the PP business system both as-is and from a process re-engineering standpoint, to understand the application domain, and to characterize end-customers of the reference organization.
(vi) Became aware of the technologies, with respect to the specific reference context and point of view.
(vii) Completed the MSS process.

 

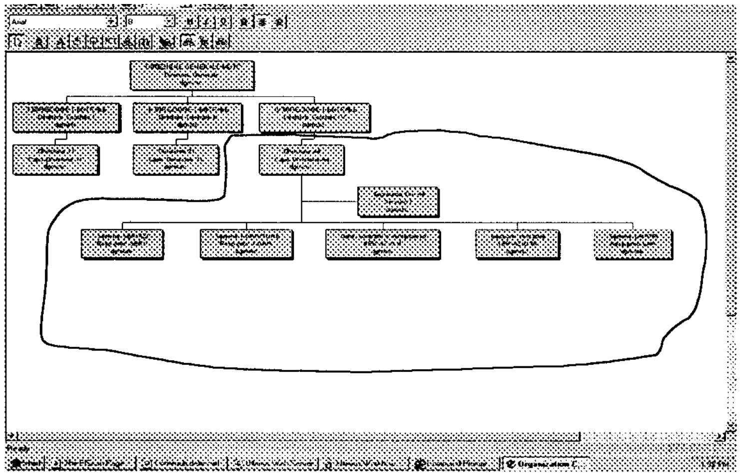

Figure 6. The organization chart of the reference organization, with the pilot agency encircled.

### 5.1. The reference organization

We present the case of a reference organization that is an organization of end-users (nominally the final destinations of the SW products). We are working to apply the same approach to SW maintainers and developers.

In particular, we consider a service organization. Let us denote such an organization as a MO. Within a MO, an organization to host a PP was chosen (point (iii) above). This is a typical medium-sized DO of a central administration. The DO is structured and behaves similarly to a large class of other administrative agencies. Figure 6 includes part of the formal organization chart of the overall organization; the pilot agency is shown by the region that is ringed. Figure 6 was developed by using the Ultimus™ workflow suite. The DO works on documents, and is aimed at acquiring furniture and services and distributing them in the overall organization. The DO manages money but does not handle it directly. Expenses and related activities are planned by year, and are described in a program that the DO arranges and the MO approves by the end of each year. While the program is static, the overall system is flexible in its support activities and it is able to handle exceptions.

### 5.2. Goals

Following the GQM (Basili, Caldiera and Rombach, 1994b; Basili and Rombach, 1988), Table II shows the structure of goals, which is based on five goal facets, plus an additional *Constraint* facet. The goals are listed in summary form only, but with some detail on *quality focuses*.

The five goal facets are:

- *Object of study*, which represents the target entity;

 

Table II. The GQM structure of the reference organization goals.

| Goal facet | Description |
|---|---|
| *Object of study* | BPR. |
| *Context* | TT laboratory, in prospective DO, and then MO. |
| *Point of view* | MO's management. |
| *Purpose* | Understand BPR pros and cons. |
| *Quality focuses* | Improve:<br>• relationships with external customers,<br>• quality of the workplace (internal customers).<br>Increase:<br>• performance of the processes,<br>• reliability of the processes,<br>• transparency of the processes,<br>• privacy of the information. |
| *Constraints* | Do not modify:<br>• current structure of the processes in place,<br>• hierarchical structure of the involved formal positions and roles.<br>Keep:<br>• innovation and related technologies under management control,<br>• costs inside a defined, acceptable market window. |

- *Context*, that is the scope of the goals;
- *Point of view*, that is the reference organization as seen by the innovation-driver; and
- *Purpose*, which registers the underlying reason, e.g. understand, improve, monitor, evaluate, predict, and change.

### 5.3. From goals and baseline to a technology domain

From the goals in Table II, and from the constraint for saving roles and processes in place (and hence, minimizing, and hopefully offsetting the dislocation of people), we derived the need to apply to the MO a minimal BPR intervention. In particular, we adopted a WA approach (point (iv) above) because WA technology is suitable for automating work processes, within stated constraints. In particular, with respect to the MO and DO, WA can help in:

(i) introducing few changes to actual paper-based processes (e.g. document scan and electronic signatures);
(ii) transferring tedious controls, rooting, storage and retrieval of documents to information technology; and
(iii) leaving the change of state of the main WA abstractions, e.g. processes, document, people, competitive tender, and work groups, under the control of the work-step owner (manager or employee).

 

Moreover, based on the reference organization's specific goals, baseline (including scarcity of IT expertise, redundancy in personnel), and needs (to train people in IT and starting mostly from the basics, to assign a well identified task to each person, and to provide ongoing training for personnel in their jobs), the organization could use computer-science concepts and WA technologies contextually to:

- advance personnel expertise, and
- improve the reliability of processes.

For instance, sequential recovery work-groups and/or multiple redundant parallel work-groups could be established organization-wide, without affecting the external conceptual view of the in-place processes: see fault-tolerant recovery-blocks, redundant executions, and cold/hot spares (Randell, 1975).

Finally, WA concepts are, and WA applications are requested to be, flexible enough to support more radical maintenance intervention in the organization's business processes, which could be required in the future.

### 5.4. Modeling the business systems and changes

We analyzed and modeled the DO business processes in the PP. The modeling activity occurred twice during the BPR life cycle (Harmon and Watson, 1997; AIPA, 1997). The first time was when we analyzed the current business process; the second time was when we identified, designed, and constructed the changes. Contextually, we identified the SPs that, in our opinion, were valid for the overall organization MO.

After a preliminary high-level analysis of the DO business processes, in order to carry out a detailed analysis, we applied a use-case driven approach. We used UML actors, use-cases, entity objects, interface objects, and control objects, as modeling tools (Harmon and Watson, 1997; Rumbaugh, Jacobson and Booch, 1999), and OSE as an analysis methodology (Jacobson *et al.*, 1993). The business model included the requirement model, the actor model, and use-case model, and the analysis model.

Based on the analysis above, we specified the PP in detail. We hence derived the LP as a large part of the PP. Table III includes an UML/OSE characterization of the LP business system model before changes and after changes, respectively. For further details about the SPs and the LP, see sub-Section 5.6.

### 5.5. Becoming aware of technologies

We started with looking at WA web pages, and discussing with technicians about the WA technology that they were aware of. Hence, based on technology documentation, we applied the on/off model and feature analysis.

During the further development of the experimentation and based on the process framework shown above, we asked the M (the measurers) to supply the E (the experimenters) with flexible models. In the beginning, both TTEF models and EIEF models were not stable or reliable, so that experimenters and TT modelers continually fed one another. Consequently, we borrowed as best we could from Cantone (1999), and attempted to build an MM-base (see Figure 2). In particular, we adopted, formalized, and, finally, specialized the CEM (Cantone, 1999) to the technology. That model, the CEM which

Table III. Characterization of the LP and the synthetic processes.

| Modeling entity | Number before WA | Number after WA |
|---|---|---|
| Systems | 1 | 1 |
| Sub-systems | 10 | 10 |
| Actors | 25 | 27 |
| • people | 21 | 22 |
| • primary | 8 | 8 |
| • secondary | 13 | 14 |
| • systems | 4 | 5 |
| Classes | 65 | 70 |
| Associations | 92 | 100 |
| • inheritance | 9 | 10 |
| • others | 83 | 90 |
| Use-cases (full courses) | 24 | 40 |
| • PP | 24 | 40 |
| • synthetic processes | 5 | 5 |

is sketched out in the appendix, is suitable for expressing the 'attitudes' of a technology to the organization's goals (*attitude* MM), and can be used to model technology costs, benefits, features, and value (*value* MM).

Initially, as regards the evaluation of benefits, we used the attitude measure of benefits to guide us towards appropriate technology, in order to provide the reference organization with an advanced and adequate WA environment. In practice, a weighted and scored technology tree with few levels of goal decomposition and basic constraints was used (see appendix).

Successively, in order to get a limited set of treatments and, among these, the MMAT, we applied the attitude MM in depth by using deep and complete technology trees (Cantone, 1999; Cantone and Donzelli, 1999).

### 5.6. Executing the experiment

Based on the available resources and on the set deadline for delivering the results and proposing a specific WA development technology to the reference organization, we moved on to develop the feasibility analysis and to chose the experimental technique to follow. We chose the MSS technique (see sub-Section 4.6).

Four SPs, ($SP_{1–4}$), we selected for multiple developments by using the control technology and MMAT. After training, which needed one more SP, ($SP_T$), two development teams (A and B) were assigned to work in parallel in real time. Each group developed an SP, with each team using competing technologies, but with no overlap between the groups in the technologies used.

 

Table IV. The allocation of groups A and B for actual parallel development, and of group C for virtual parallel development, of synthetic processes.

| Phase | Step | Control group | Activity | Treatment group | Activity | Synthetic process |
|---|---|---|---|---|---|---|
| 0 | 0.1 | A, B, C | 0.1.1 | A, B, C | 0.1.2 | Common training |
| | 0.2 | A, B | 0.2.1 | A, B | 0.2.2 | $SP_T$ |
| | | C | 0.2.3 | C | 0.2.4 | Training |
| 1 | 1.1 | B | 1.1.1 | A | 1.1.2 | $SP_1$ |
| | | C | 1.1.3.$n$ | C | 1.1.4.$n$ | $SP_3$ |
| 2 | 2.1 | A | 2.1.1 | B | 2.1.2 | $SP_2$ |
| | | C | 2.1.3.$n$ | C | 2.1.4.$n$ | $SP_4$ |
| 3 | 3.1 | A | 3.1.1 | — | — | $SP_3$ |
| | | — | — | B | 3.1.2 | $SP_4$ |
| | | C | 3.1.3.$n$ | C | 3.1.4.$n$ | $SP_2$ |
| 4 | 4.1 | C | 4.1.1.$n$ | C | 4.1.2.$n$ | $SP_1$ |

Moreover, one additional group (C), a synthetic group, was appointed to develop SPs in virtual parallelism. Starting from one of the not-yet allocated SPs, the group was instructed to alternate the use of the technologies every morning, and use a technology for four hours in a day (a time-quantum).

SP developments were designed to evolve as in Table IV, where both phases, and their steps, are intended to proceed in sequences, while step activities can proceed in any order, hopefully in parallel, whether allocated to diverse groups or not. Note that '$n$' denotes an activity-quantum (virtual parallelism), and proceeds in the series of the positive integer numbers, up to the conclusion of the associated developments.

The experimentation summarized in Table IV was also intended to compare the performance of virtual and real parallelism in SW experimentation. Unfortunately, after some time from the beginning of the experiment, the resource owner claimed 'redundant' resources. Only the timeshared group (C) continued to work. As an instance of such developments, Figure 7 shows a critical path process map of the development of a synthetic process, which was targeted to manage formal competitive tenders in the context of Italy's PA. Figure 7 was developed by using the Ultimus$^{TM}$ workflow suite.

All of the synthetic developments were observed, data were collected, empirical evaluations were successfully carried out, and the EMAT was detected. After EMAT and MMAT were verified as being the same, the dimensions of the PP were scaled down for laboratory development by the most adequate technology only. In practice, the general architecture of the pilot was kept, but the requirements on performance, user-interface quality, minimal number of work places, etc. were left out of the laboratory PP. The LP development process was started and observed. The collected data were successfully compared with those obtained from the previous phases. This provided us with empirical evidence of the validity of the proposed process framework and the architecture for running TT experiments. As a side effect, further accreditation was given to M for the TT attitude MM. The data in Table III also

 

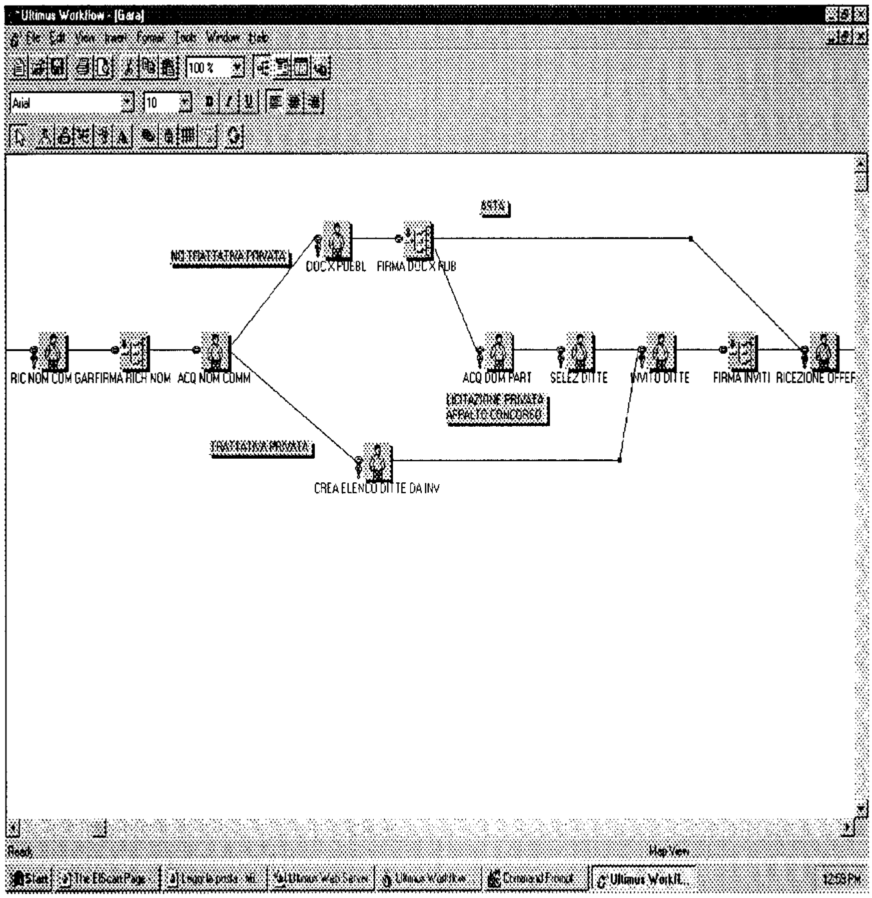


Figure 7. The process map of one of the implemented synthetic processes: managing competitive tenders (in Italian).

summarizes the development work. Further evidence and accreditation are expected from the inaction in the field of the PP.

As expected, the experiment proved to be affordable, but relatively expensive and time consuming. The experimental work needed a cumulative effort of about fifteen calendar months spread over a year and a half. The author and an undergraduate student in his last year did the work, with the help of some professional technicians. The team included an expert on the PP end-user organization (Cantone and Fusco, 1999). All of the people worked part-time on the project, and all, except the student, worked as unpaid volunteers. Table V describes the distribution of working time among some of the roles involved.

 

Table V. The roles involved in the case study.

| Role | Effort (man-months) |
|---|---|
| Project manager | 0.8 |
| Business process analyst | 3.5 |
| Customer organization | 0.4 |
| Reference organization | 0.3 |
| WA developer and measurer | 5.6 |
| WA application verifier | 1.4 |
| Inspector | 0.9 |
| Observer and editor | 1.4 |

### 5.7. Lessons learned

The lessons that we learned, while applying the MSS evaluation process model focused mainly on project management, which include the following.

(i) Never start multiple developments (or even experiments) of a synthetic process in parallel without having the manager of the people being strongly committed. As a backup, be ready at anytime to switch from real to virtual parallel developments. Keep the assignment of a virtual parallel development to a synthetic group as a fallback possibility from the beginning of the experiment.

(ii) Do not start a TT experiment without first having a strongly-committed measure modeler. Ensure continual management and research leadership support to experimenters.

(iii) When a well developed MM gives more than three acceptable treatments, the model granularity is either too large for the case, the model is wrong, you applied it incorrectly, or the investigated technology is more usual (and imitated) than advanced.

## 6. CONCLUSIONS AND FUTURE WORK

This paper proposed process framework and conceptual architecture for enacting TT empirical investigations. Goal-driven MMs, such as developed by the TTEF, drive the process framework. The proposed process and architecture, in their turn, are entities of the EIEF.

The proposed framework was developed and verified by the Experimental Software Engineering group of the University of Rome at Tor Vergata. The group developed multiple synthetic processes and an extended use-case-based analysis, in the domain of the workflow automation of business processes, in the context of a medium-sized administrative agency, which is an exemplar of a large class of service organizations.

Further work is planned to apply the proposed framework to field projects. Such work is also expected to take into full consideration the value of a technology, in addition to its benefits and features,

from each of the points of view of the organizations that are involved with the TT, such as maintainers, developers and end-users.

## APPENDIX: Applying a 'Common Evaluation Model' for evaluating the utility of a TT empirical investigation

The common evaluation model (CEM) (Cantone, 1999) is a goal-driven model for evaluating technology-related values. The word 'Common' in its name refers to its intended range of applicability to:

- modeling any technology-related entity, and
- evaluating that entity's desired/expected/actual/prospective short-range or long-range costs, benefits, performances, features, or values, and related ratios or differences, and trends.

An example follows, which sketches a use of a simplified version of the CEM to weigh running a TT empirical investigation versus relying on using available models. Four tables summarize this case of evaluating the utility of a technology experiment for an organization, see Tables VI–IX. These tables are representations of a WST that is a 'whole-part' hierarchy of the attributes that affect the technology experiment. See also the final example in Cantone and Donzelli (1999) for a short description of the CEM in a context of technology-benefits evaluation.

In a WST, a *weight* represents the relative relevance of the corresponding attribute for the organization. Any node inherits weight from its parent, proportionally to its importance with respect to the sibling nodes (see local percentages in Tables VI–IX). The *score* of a leaf is a measure of the leaf-attribute.

Tables VI and VII show 'hybrid' WSTs. They include costs and other kinds of attributes. The weight of the root is constrained to 1.0, and other weights are consequently assigned. Hence, all node weights are greater than 0.0 (the tree rooted in such an unimportant 0.0 node should be erased), and the weights of all of the nodes less the root are less than 1.0 (such a 1.0 node should be the same as that of its father-node). Scores are scaled 0.0–1.0, where, for a leaf–node, 1.0 (respectively 0.0) denotes the maximum (minimum) of the attribute, which could be reasonably included by a technology. The method used to score a node, $n$, is to set Score ($n$) as in the following:

$$\text{Score}(n) = \sum_{\forall L \epsilon H(n)} \text{Score}(L) * \text{Weight}(L) \qquad \text{(A1)}$$

where $H(n)$ is any consistent and complete set of heirs of $n$, e.g. the set of leaves descending from $n$. Consequently, Score (Root) holds 1.0 (respectively 0.0), if and only if all the remaining nodes are set to 1.0 (respectively 0.0) and *vice versa*.

Table VI shows WST #1 as an instance of the attributes that push an organization to run a TT empirical-evaluation, rather than applying some available models. The greater the weight and score of an attribute, the greater the attribute contribution to sustain experimentation as an opportunity for the organization. In contrast, Table VII shows attributes that do not support the running of an experimentation. Table VIII shows some constraints that are associated with the empirical evaluation process.

 

Table VI. The technology tree, and the aptitude of an empirical evaluation.

| Attribute | Weight Global (Local %) | Score | Aptitude |
|---|---|---|---|
| Attribute tree #1 | 1.000 (100%) | 0.396 | 0.396 |
| • Uncertainty of the technology need | 0.150 (15%) | 0.800 | 0.120 |
| • Newness of the technology: | 0.160 (16%) | 0.519 | 0.083 |
| ◆ for the market | 0.112 (70%) | 0.630 | 0.070 |
| ◆ for the organization | 0.048 (30%) | 0.270 | 0.013 |
| • Cost of the technology: | 0.220 (22%) | 0.261 | 0.058 |
| ◆ acquisition | 0.022 (10%) | 0.380 | 0.023 |
| ◆ lost alternative opportunities | 0.022 (10%) | 0.048 | 0.003 |
| ◆ training | 0.132 (60%) | 0.285 | 0.017 |
| ◆ technology upgrading | 0.044 (20%) | 0.237 | 0.015 |
| • Impacts of the technology on the organization | 0.200 (20%) | 0.130 | 0.026 |
| ◆ monetary impact | 0.010 (5%) | 0.070 | 0.070 |
| ◆ organizational impact | 0.060 (30%) | 0.350 | 0.021 |
| ◆ social impact | 0.080 (40%) | 0.105 | 0.084 |
| ◆ impact on knowhow | 0.050 (25%) | 0.175 | 0.087 |
| • Time-window available to introduce the technology and use competitive advantages | 0.070 (7%) | 0.200 | 0.014 |
| • Organization dimension to technology-supplier dimension ratio | 0.050 (10%) | 0.500 | 0.025 |
| • Organization's technology leadership (versus followership) | 0.050 (10%) | 0.010 | 0.000 |
| • Simplicity of the experiment: | 0.100 (10%) | 0.750 | 0.075 |
| ◆ experimental maturity | 0.050 (50%) | 0.600 | 0.030 |
| ◆ expected validity of synthetic processes | 0.050 (50%) | 0.900 | 0.045 |

Table VII. The technology tree and the inaptitude of an empirical evaluation (versus evaluation models).

| Attribute | Weight Global (local %) | Score | Inaptitude |
|---|---|---|---|
| Attribute tree #2 | 1.000 | 0.312 | −0.312 |
| • Similarity of the new technology with the ones already in use in the organization | 0.050 (5%) | 0.010 | 0.000 |
| • Availability and applicability of the new technology and availability and reliability of technology evaluation models | 0.500 (40%) | 0.500 | −0.250 |
| • Cost of the experiment | 0.050 (5%) | 0.100 | −0.005 |
| • Duration of the experiment | 0.100 (10%) | 0.500 | −0.050 |
| • Risk the organization is ready to take as an effect of introducing new technology | 0.125 (12.5%) | 0.010 | −0.001 |
| • Knowledge the organization has about the technology supplier | 0.125 (12.5%) | 0.050 | −0.006 |
| • Scarcity of financial resources | 0.050 (5%) | 0.010 | 0.000 |

Table VIII. The constraints of the empirical evaluation.

| Attribute | Constraint |
|---|---|
| The technology is available and applicable | On/Off |
| The cost of the experiment | <Euro 8000 |
| The duration of the experiment | <16 months |
| An independent research laboratory (innovation-driver) is available to run the experiment under the constraints above | On/Off |

Table IX. The utility of an empirical evaluation for an organization.

| | |
|---|---|
| Utility rate ($U_R$) | 1.269 |
| Utility sigma ($U_S$) | +0.084 |

Measures can be defined on a WST. For instance, *Aptitude* (*Inaptitude*) of the experiment in meeting organization goals can be defined as Score (Root) (−Score (Root)) see Table VI (Table VII), right-most column.

Note that the method described by expression (A2) could be used to score a 'homogeneous' WST, e.g. a cost WST (Cantone, 1999):

$$\text{Score }(n) = \sum_{\forall L \in D(n)} \text{Score}(L) \tag{A2}$$

where $D(n)$ is the set of the children of $n$.

The measures can be also defined on a system of WSTs. For instance, with respect to examples above, Table IX shows the *utility of the empirical evaluation (U)*, which is defined as a rate $(U_R)$ by expression (A3) and a summation ($U_S$) by expression (A4).

$$U_R = \text{Aptitude (WST \#1)}/\text{Aptitude (WST \#2)} \tag{A3}$$

$$U_S = \text{Aptitude (WST \#1)} + \text{Inaptitude (WST \#1)}. \tag{A4}$$

**ACKNOWLEDGEMENTS**

The author thanks Professor Luigi Cantone of the University of Naples for insightful discussions about the technology transfer system. He also thanks Dr. Paolo Donzelli, presently with RMCS Shrivenham College at Cranfield University, for his constructive comments and paper review, and to Ing. Fabrizio Fusco, of the Italy Central Administration, for his collaboration in the analysis and construction work. Thanks are also due to Eustema SpA Rome, and to the staff of the Administration and Documentation Computing Center of the University of Rome

 

at Tor Vergata, for tool-support and assistance. Finally, the author thanks the *Journal's* anonymous reviewers for their very detailed analyses and comments on the paper.

**REFERENCES**

AIPA 1997. *Reingegnerizzazione dei processi e impiego delle tecnologie: due modalità da integrare per il cambiamento, Bollettino 1, 1997*; Autorita' Informatica per la Publica Amministrazione: Roma, Italy.

Basili VR. 1989. The experience factory: packaging software experiences. *Proceedings of the NASA Goddard Space Flight Center's 14th Annual Software Engineering Workshop*; National Aeronautics and Space Administration GSFC: Greenbelt MD; pp. 12–34.

Basili VR. 1992. The experimental paradigm in software engineering. *Lecture Notes in Computer Science*, vol. 706; Springer-Verlag: Berlin; pp. 3–12.

Basili VR. 1993. The maturing of the quality improvement paradigm in the SEL and experience factory fundamentals. *Proceedings of the NASA Goddard Space Flight Center's 18th Software Engineering Workshop*; National Aeronautics and Space Administration GSFC: Greenbelt MD; pp. 39–48.

Basili VR. 1996. The role of experimentation in software engineering: past, current and future. *Proceedings 18th International Conference on Software Engineering*; IEEE Computer Society Press: Los Alamitos CA; pp. 442–449.

Basili VR, Caldiera G, Cantone G. 1992. A reference architecture for the component factory. *ACM Transactions on Software Engineering and Methodology* **1**(1):53–80.

Basili VR, Caldiera G, Rombach HD. 1994a. The experience factory. *Encyclopedia of Software Engineering*; John Wiley & Sons, Inc.: New York NY; pp. 469–476.

Basili VR, Caldiera G, Rombach HD. 1994b. The goal question metric approach. *Encyclopedia of Software Engineering*; John Wiley & Sons, Inc.: New York NY; pp. 528–532.

Basili VR, Green S, Laitemberg O, Lanubile F, Shull F, Sorumgard S, Zelkowitz MV. 1996. The empirical investigation of perspective-based reading. *The Empirical Software Engineering Journal* **1**(2):133–164.

Basili VR, Rombach HD. 1988. The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* **14**(6):758–773.

Basili VR, Selby RW, Hutchens DH. 1986. Experimentation in software engineering. *IEEE Transactions on Software Engineering* **12**(7):773–743.

Berztiss A. 1996. *Software Methods for Business Reengineering*; Springer-Verlag: New York NY.

Briand L, Bunse L, Daly J, Differding C. 1997. An experimental comparison of the maintainability of OO and structured design documents. *The Empirical Software Engineering Journal* **2**(3):291–312.

Brown AW, Wallnau KC. 1996. A framework for evaluating software technology. *IEEE Software* **13**(9):39–49.

Burgess A. 1995. Mad about or mad at measurement? *IEEE Software* **12**(1):115–116.

Cantone G. 1999. *Software Technology Evaluation Models*, Technical Report; University of Rome at Tor Vergata: Roma, Italy.

Cantone G, Donzelli P. 1998. Production and maintenance of software measurement models. *Project Control for 2000 and Beyond*; Shaker Publishing BV: Maastricht, The Netherlands; pp. 51–58.

Cantone G, Donzelli P. 1999. Goal-oriented software measurement models. *Project and Control for Software Quality*; Shaker Publishing BV: Maastricht, The Netherlands; 355–364.

Cantone G, Fusco F. 1999. *Una Sperimentazione di Automazione dei Flussi di Lavoro per un Dipartimento della Publica Amministrazione—Processi Sintetici e Progetto Pilota*; University of Rome at Tor Vergata: Roma, Italy.

Ciolkowsky M, Differding C, Laitemberg O, Munch J. 1997. *Empirical Investigation of Perspective-based Reading: a Replicated Experiment*, ISERN Technical Report 97-13; Fraunhofer Institute for Experimental Software Engineering, University of Kaiserlautern: Kaiserlautern, Germany.

ESI page. 1999. http://www.esi.es/publications/reports/techplan.pdf [16 October 1999].

Garvin DA. 1993. Building a learning organization. *Harvard Business Review.* **71**(4):78–91.

Hammer M, Champy J. 1995. *Reengineering the Corporation: a Manifesto for Business Revolution*. Nicholas Brealey Publishers: London, UK

Harmon P, Watson M. 1997. *Understanding UML: The Developer's Guide with a Web-based Application in Java*™; Morgan Kaufmann Publishers, Inc.: San Francisco CA

ISERN page. 1999. http://www.iesi.fhg.de/ISERN/pub/isern_biblio_tech.html [18 October 1999].

Jacobson I, Christerson M, Jonsson P, Overgaard G. 1993. *Object-oriented Software Engineering: A Use Case Driven Approach*; Addison-Wesley: Essex, UK

Kamstie E, Lott CM. 1995. *An Empirical Evaluation of Three Defect-Detection Techniques*; ISERN Technical Report 95–02; Fraunhofer Institute for Experimental Software Engineering, University of Kaiserlautern: Kaiserlautern, Germany.

Kitchenham BA. 1996a. *DESMET: a Method for Evaluating Software Engineering Methods and Tools*, Technical Report 96-09; Department of Computer Science, University of Keele: Staffordshire, UK.

 

Kitchenham BA. 1996b. Evaluating software engineering methods and tool (part 1), *ACM SIGSOFT Notes* 1996b**21**(1):11–15; 1998c**21**(2):11–15; 1997**22**(5):10–12.

Kitchenham BA, Pickers L, Pfleeger SL. 1995. Case studies for method and tool evaluation. *IEEE Software* **12**(4):52–62.

Ministero. 1998. *Materiali per un Piano d'Azione* (draft). Ministero delle Communicazioni: Rome, Italy.

Norman R, Ramierez R. 1993. Designing interactive strategies: from value chain to value constellation. *Harvard Business Review* **71**(4):65–77.

Pfleeger SL. 1994. Experimental design and analysis in software engineering: part 1. *ACM SIGSOFT Notes* 1994**19**(4):16–20; 1995a**20**(1):22–26; 1995b**20**(2):14–16; 1995c**20**(3):13–15; 1995d**20**(5):14–17.

Porter A, Votta LG, Perry D, Staudenmayer N. 1995. *Experimental Software Engineering: a Report on The State of the Art*; Sloan School of Management, Massachusetts Institute of Technology: Cambridge MA.

PITAC. 1998. *Interim Report to the President*; President's Information Technology Advisory Committee, National Coordination Office for Computing, Information, and Communication: Arlington VA.

Premier Ministre. 1998. *Preparér l'entrée de la France dans la société de l'inform@ation—programme d'action gouvernamental*; La Documentation Francaise, (1), Government of France: Paris, France.

Premier Ministre. 1999. Société de l'information: discours du Premier ministre à l'Université d'été de la communication Hourtin, jeudi 26 août 1999: http://www.internet.gouv.fr/francais/index.html [18 October 1999].

Prime Minister's Office of the Hellenic Republic. 1999. Greece in the information society: strategy and actions (draft). The Prime Minister's Office, Government of Greece: Athens, Greece. URL: Primeminister.gr/infosoc/index_en.htm [16 October 1999].

Randell B. 1975. System structure for software fault-tolerance. *IEEE Transactions on Software Engineering* **1**(2):220–232.

Rombach HD, Basili VR, Selby RW (eds). 1992. Experimental software engineering issues: a critical assessment and future directions. *Lecture Notes in Computer Science*, vol. 706; Springer-Verlag: Berlin, Germany.

Rumbaugh I, Jacobson I, Booch G. 1999. *The Unified Modeling Language Reference Manual*; Addison-Wesley: Reading MA.

Van Solingen R, Berghout E. 1999. *The Goal/Question/Metric Method*; McGraw-Hill: Maidenhead, UK.

Visaggio G. 1997. Assessing the maintenance process through replicated, controlled experiments. *The Journal of Systems and Software* **44**(3):187–197.

Zelkowitz MV. 1994. Algebra and Models (and Reality). *ACM SIGSOFT Notes* **19**(4):79–81.

Zelkowitz WV, Wallace D. 1998. *Validating the Benefits of New Software Technology (draft)*; Computer Science Department, University of Maryland: College Park MD.

## AUTHOR'S BIOGRAPHY

**Giovanni Cantone** is an Associate Professor of Computer Science in the Department of Informatics, Systems and Production Management of the Faculty of Engineering, and a member of the Scientific–Technical Committee of the Administration and Documentation Computing Center of the University of Rome at Tor Vergata, Roma, Italy. He has also served at the University of Naples and the University of Maryland. Giovanni is a founding member of the International Software Engineering Research Network, an affiliated member of the UMD Experimental Software Engineering Group, and has been on the National Board of the Italian Association of Informatics and Automatic Computing. Since 1974, he has carried out software engineering research. His current research interests include software maintenance processes, process re-engineering, software technology transfer, software measurements, empirical software engineering, and software quality. His doctoral 'Laurea' degree is from the Faculty of Engineering of the University of Naples. His email address is: cantone#@info.uniroma2.it